Dynamic Topology-Aware Flow Path Construction and Scheduling Optimization for Multilayered Continuous-Flow Microfluidic Biochips

Meng Lian
Technical University of Munich
Munich, Germany
m.lian@tum.de

Shucheng Yang
Technical University of Munich
Munich, Germany
shucheng.yang@tum.de

Mengchu Li
Technical University of Munich
Munich, Germany
mengchu.li@tum.de

Tsun-Ming Tseng
Technical University of Munich
Munich, Germany
tsun-ming.tseng@tum.de

Ulf Schlichtmann
Technical University of Munich
Munich, Germany
ulf.schlichtmann@tum.de

ABSTRACT

Multilayered continuous-flow microfluidic biochips are highly valued for their miniaturization and high bio-application throughput. However, challenges arise as the dynamic connections of channels, adjusted to satisfy varying demands of fluid transportation at different moments, complicate the execution of bioassays. The existing methods often focus on device binding and operation scheduling during high-level synthesis but overlook the topological connections within the microfluidic network. This oversight leads to mismanagement of conflicts between fluid transportations and erroneous assumptions about constant flow velocities, resulting in decreased accuracy and efficiency or even infeasibility of bioassay execution. To address this problem, we mathematically model the flow velocity that varies according to the dynamic changes of the topological connections between the on-chip components during the execution of the bioassay. Further integrating the flow velocity model into the high-level synthesis, we propose a quadratic programming (QP) method that constructs flow paths and optimizes scheduling schemes to minimize the bioassay completion time. Experimental results confirm that, compared with the state-of-the-art approach, our method shortened the bioassay completion time by an average of 40.9%.

KEYWORDS

Multilayered continuous-flow microfluidic biochip, High-level synthesis, Quadratic programming

ACM Reference Format:
Meng Lian, Shucheng Yang, Mengchu Li, Tsun-Ming Tseng, and Ulf Schlichtmann. 2025. Dynamic Topology-Aware Flow Path Construction and Scheduling Optimization for Multilayered Continuous-Flow Microfluidic Biochips. In *30th Asia and South Pacific Design Automation Conference (ASPDAC '25), January 20–23, 2025, Tokyo, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3658617.3697779

ASPDAC '25, January 20–23, 2025, Tokyo, Japan

ACM ISBN 979-8-4007-0635-6/25/01.
https://doi.org/10.1145/3658617.3697779

1 INTRODUCTION

Multilayered continuous-flow microfluidic biochips (mCFMBs), also known as lab-on-a-chip systems, have gained widespread adoption in recent years as a promising platform for high-throughput biological applications. The mCFMBs facilitate the automatic and concurrent execution of a variety of complex assays, including DNA purification [1], environment monitoring [2], and cell culture [3].

The construction of mCFMBs utilizes the soft lithography technique to bond multiple patterned elastomer layers, each consisting of dedicated channels that allow gas or fluids to pass through [4]. The typical configuration of mCFMBs features a *flow layer* and a *control layer*. Reactants or reagents, imported from external instruments to the flow channels, are processed by manipulating the pressure within the control channels, enabling various biomedical and biochemical operations like mixing, heating, filtering, and detection [5].

The challenges of synthesizing mCFMBs include *high-level synthesis*, which involves binding biochemical operations to devices and scheduling each operation by determining its start and end times, and *physical-level synthesis*, which focuses on device placement and channel routing. As the complexity and scale of integration in mCFMBs expand, the manual design process becomes increasingly time-consuming and error-prone, requiring automated synthesis tools to realize the bioassay on a feasible and optimized chip design with an execution protocol. Significant efforts have been invested in recent years to develop comprehensive design automation solutions. Tseng *et al.* [6] proposed a top-down synthesis method, which enhances resource binding and placement to reduce valve-switching and bioassay completion times. Yao *et al.* [7] proposed a flow-control codesign methodology to enhance the design quality of mCFMBs by seamlessly combining flow-layer and control-layer design stages. Li *et al.* [8, 9] proposed high-level modeling methods to optimize resource utilization according to specified application protocols. Minhass *et al.* [10, 11] proposed scheduling and fluid routing approaches that map operations to given physical topologies. Tseng *et al.* [5, 12, 13] proposed place-and-route tools

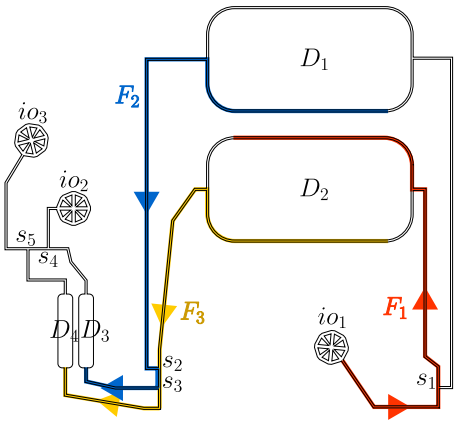


Figure 1: Illustration of a partial biochip synthesized using Columba 2.0 [5], where io_1–io_3, s_1–s_3, D_1–D_4, and F_1–F_3 denote flow ports, flow channel branches, devices, and fluid transportations, respectively.

that generate physical designs ready for manufacturing, capable of supporting applications of varying scales. Li *et al.* [4] proposed a simulation-based approach to construct valid fluid paths for given chip designs.

Though the proposed approaches gradually progress toward a comprehensive automatic synthesis flow, divergences between the chip design and execution protocols continue. These divergences are primarily attributed to the dynamic connections of channels, which result from selectively allowing or blocking channels to form different flow paths that meet changing operational demands. The following considers the flow-layer structure in Figure 1 to analyze the limitations and drawbacks suffered by the existing methods.

Firstly, existing high-level synthesis methods often inadequately address *conflicts* arising when multiple fluid transportations simultaneously traverse through shared vertices such as inlets, outlets, devices, or channel branches. These conflicts can contaminate reactants and reagents or cause unexpected channel blockages. During fluid transportation, each flow path starts with an inlet connected to an external pressure source to facilitate fluid movement and ends with an outlet to release air and prevent blockages [4]. For example, consider three fluid transportations, F_1, F_2, and F_3, transport fluids from io_1 to D_2, D_1 to D_3, and D_2 to D_4 with flow paths $P_1 = (\boldsymbol{io_1}, \boldsymbol{s_1}, \boldsymbol{D_2}, s_2, s_3, D_3, s_4, io_2)$, $P_2 = (io_1, s_1, \boldsymbol{D_1}, \boldsymbol{s_2}, \boldsymbol{s_3}, \boldsymbol{D_3}, s_4, s_5, io_3)$, and $P_3 = (io_1, s_1, \boldsymbol{D_2}, \boldsymbol{s_2}, \boldsymbol{s_3}, \boldsymbol{D_4}, s_5, s_4, io_2)$, respectively. The following illustrates conflicts that can arise in fluid transportation:

- Existing conflict identification [10] primarily focuses on conflicts arising from fluid transportations where the transported fluids traverse common vertices. However, the interactions between F_1 and F_2, which transport fluids through non-intersection paths, reveal an overlooked conflict. Specifically, executing F_1 requires blocking the channel between s_1 and D_1. This blockage is necessary because s_1 is a channel branch; without it, the fluid could mistakenly flow to D_1. However, this manipulation disconnects F_2 from its inlet and removes the necessary pressure to enable fluid movement in F_2. In other words, F_1 and F_2 cannot be executed in parallel, or conflict may occur.
- Contrary to the constraints introduced in [10] that flow paths with common vertices cannot be executed in parallel, the flow paths P_1 and P_3, which share the subpaths $io_1 \rightarrow s_3$ and $s_4 \rightarrow io_2$, demonstrate an exception. Here, fluids within F_1 and F_3 do not encounter each other; instead, only air or buffer liquid traverses these shared subpaths, presenting no risk of contamination. Consequently, F_1 and F_3 can be executed in parallel without conflict.
- Existing methods often resolve conflicts by executing fluid transportations one after the other's completion [10], which can unnecessarily prolong the bioassay completion time. Consider F_2 and F_3, which exhibit a conflict due to transporting different fluids along the shared subpath $s_2 \rightarrow s_3$. Instead of executing them one after the other, an optimal execution would only involve fluids within F_2 and F_3 sequentially traversing s_2 and s_3 to minimize delays.

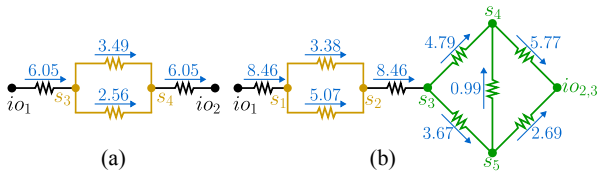


Figure 2: Equivalent fluid circuits (a) $c(P_1, P_3)$ and (b) $c(P_1, P_2, P_3)$, with flow velocities (in $\mathrm{mm\,s^{-1}}$) indicated in blue, and the parallel and bridge connections in yellow and green, respectively.

Secondly, existing high-level synthesis methods often overly simplify the microfluidic network by assuming constant fluid transport latencies and standardized flow velocities, typically fixed at a constant of $10\,\mathrm{mm\,s^{-1}}$ [10, 11]. However, the microfluidic network includes non-serial connections that result in significantly varied flow velocities across different channels. For example, when executing F_1 and F_3 in parallel, the resulting equivalent fluidic circuit $c(P_1, P_3)$, as shown in Figure 2(a), demonstrates a *parallel* connection, highlighted in yellow in Figure 2(a). Moreover, the fluid circuit combining P_1, P_2, and P_3 contains a *bridge* connection, highlighted in green in Figure 2(b), which has never been addressed in existing methods. Figure 2 also illustrates flow velocities and directions for channel segments, indicated in blue, which vary significantly. Notably, some velocities are markedly lower than $10\,\mathrm{mm\,s^{-1}}$. Further, comparing $c(P_1, P_2, P_3)$ with $c(P_1, P_3)$, the former showcases greater main flow velocities despite containing more hydraulic resistors[1]. This demonstrates the non-intuitive effect of non-serial connections where increased channel involvement does not necessarily slow the flow velocity.

Last but not least, the existing high-level synthesis methods overlook the dynamic topological changes within the microfluidic network. Specifically, modifications in channel connectivity, either

[1]The application of circuit methods to microfluidics is based on the analogous behavior of hydraulic and electric circuits, where pressure corresponds to voltage, volumetric flow rate to current, and hydraulic resistance to electric resistance. This analogy, supported by *Hagen-Poiseuille's law* akin to *Ohm's law*, assumes that the flow is laminar, viscous, and incompressible [14].

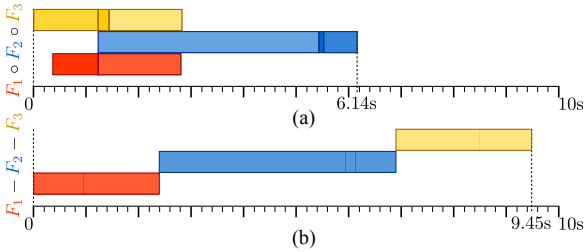


Figure 3: Scheduling schemes for executions (a) $F_1 \circ F_2 \circ F_3$ and (b) F_1–F_2–F_3.

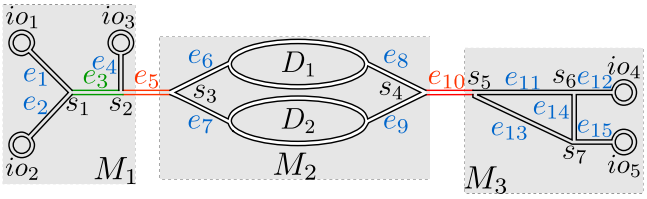


Figure 4: The weighted graph $\mathcal{A}$ with root modules M_1–M_3.

by blocking or allowing flow to meet fluid transportation demands, significantly impact flow velocities and thus affect the execution schedule. For example, in the parallel execution of F_1, F_2, and F_3, denoted by $F_1 \circ F_2 \circ F_3$, F_1 and F_3 initiate first due to a conflict with F_2, using io_2 as their outlet and forming the fluid circuit $c(P_1, P_3)$. Once the fluid within F_1 has traversed the critical vertex s_1, F_2 starts, and P_2 integrates into the fluid circuit as $c(P_1, P_2, P_3)$. This transition, illustrated in Figures 2(a) to 2(b), demonstrates dynamic flow velocity changes due to changes in topological connections.

Further, we illustrate the scheduling schemes for parallel execution $F_1 \circ F_2 \circ F_3$ and sequential execution F_1–F_2–F_3 in Figure 3, where darker colors indicate higher flow velocities. In the parallel execution shown in Figure 3(a), flow velocities dynamically fluctuate within each fluid transportation at various moments. Conversely, the scheduling scheme in Figure 3(b) for the sequential execution reveals that flow velocities are consistently uniform due to the serial connections of channels within individual paths. In particular, the parallel execution of F_1, F_2, and F_3 reduced the completion time by 35.0% relative to their sequential execution. Thus, parallel execution in fluid transportation emphasizes a great optimization potential that can significantly shorten the bioassay completion time.

This work aims to realize the bioassay with minimized completion time by proposing a quadratic programming (QP) method. Our method inputs a chip design and a bioassay with a binding function then constructs flow paths and optimizes scheduling schemes. The main contributions of our work are summarized as follows:

- It proposes a mathematical model that accurately computes flow velocities considering topological connections within the microfluidic network, specifically addressing previously overlooked bridge connections.
- It integrates the flow velocity model into the high-level synthesis, precisely identifying and resolving conflicts between fluid transportations and operations.
- It is the first work that constructs flow paths and optimizes scheduling schemes considering the interactions among concurrently executed fluid transportations and operations as well as the effects on the flow velocities.

2 PRELIMINARIES

2.1 Problem Formulation

This work aims to solve the following problem:
Inputs: A chip design and a bioassay with a binding function.

- *Chip design interpretation:* Our method interprets the flow layer structure of the design into a weighted graph $\mathcal{A}(\mathcal{N}, \mathcal{E})$, as shown in Figure 4. Here, $\mathcal{N}$ includes flow ports $\mathcal{P}$, devices $\mathcal{D}$, and channel branches $\mathcal{S}$. The edge set $\mathcal{E}$ consists of channel segments, each denoted by $e_{(n,n')}$ connecting vertices n and n' in $\mathcal{N}$, with each segment's length quantified by the weight coefficient $l_{e_{(n,n')}}$.
- *Bioassay interpretation:* The bioassay is modeled using a sequencing graph [15] $\mathcal{G}(\mathcal{V}, \mathcal{F})$, where $\mathcal{V}$ includes specified inlets and outlets $\mathcal{B}$, each specifically designated for importing reaction samples and reagents or exporting reaction products and waste, with operations $\mathcal{O}$ that can be bound to devices via the binding function. Each operation O_i has a weight c_i, representing its required execution time. The edge set $\mathcal{F}$ consists of fluid transportations F_i.

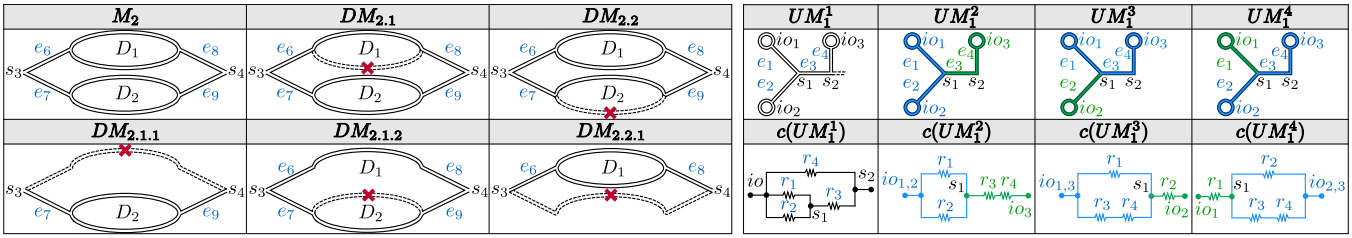
Outputs: The optimized scheduling schemes of the fluid transportations and the operations.
Subject to: The flow paths must be valid and supported by the given chip design. The parallel execution of fluid transportations must not result in conflicts.
Objective: Minimize the bioassay completion time.

2.2 Fluidic Module Construction

We partition weighted graph $\mathcal{A}$ into serially connected subgraphs to identify non-serial configurations that may arise during bioassay execution for flow velocity calculation. A subgraph of $\mathcal{A}$ that contains non-serial connections is defined as a *module*, while those derived directly from the partition of $\mathcal{A}$ are termed *root modules*. To partition $\mathcal{A}$, we remove all flow ports and incident edges from $\mathcal{A}$ and eliminate the *cut edges* [16], such as edges e_3, e_5, and e_{10} in Figure 4. The remaining subgraphs are externally connected in series through these cut edges. Next, we restore all removed flow ports and incident edges to the graph. Considering that flow ports serving as inlets (or outlets) are considered equipotential, two subgraphs connected by a cut edge still form a non-serial connection if both contain flow ports serving as inlets (or outlets). To address this problem, we merge adjacent subgraphs that include flow ports by relinking their corresponding cut edges, such as edge e_3.

We introduce the following iterative algorithm to construct all potential modules. The module construction starts with the root modules, such as M_1, M_2, and M_3 in Figure 4, and involves two processes: *degeneration* and *equipotential partition*, which are detailed as follows:

- *Degeneration:* Figure 5(a) shows that each module is degenerated by sequentially removing edges to explore new non-serial configurations. For example, M_2 is degenerated into



(a) (b)

Figure 5: Illustration of the fluid module construction, where r_i denotes the hydraulic resistance of edge e_i. (a) Degeneration of M_2. (b) Equipotential partition of M_1.

first-oder module $DM_{2.1}$ and $DM_{2.2}$, then into second-order $DM_{2.1.1}$, $DM_{2.1.2}$, and $DM_{2.2.1}$. This iterative degeneration continues until removing additional edges does not yield new non-serial configurations.

- *Equipotential partition:* Different equipotential alignments within flow-ports-equipped modules, such as M_1, are treated as new modules. For example, UM_1^1 treats ports io_1–io_3 uniformly, creating connections to other modules via s_2, as shown in $c(UM_1^1)$ in Figure 5(b). Further, M_1 also derives modules UM_1^2–UM_1^4 in Figure 5(b); these modules operate independently with their inlets and outlets, allowing reaction samples and reagents to flow directly through without serial integration with other modules. UM_1^1 – UM_1^4 will further degenerate to construct new modules.

3 MATHEMATICAL MODEL

We develop the following QP method to construct flow paths and optimize the scheduling schemes to achieve minimized bioassay completion time. The frequently used variables of our method are outlined in Table 1.

3.1 Flow Path Construction

We introduce the following constraints to ensure that flow paths are constructed validly and are supported by the given flow-layer structure.

3.1.1 Inlets and Outlets. Each flow path, denoted by P_i for F_i, must include an inlet and an outlet to create a pressure difference for fluid transportation.

$$\sum_{n\in\mathcal{P}} q^n_{i,\text{in}} = 1, \quad \sum_{n\in\mathcal{P}} q^n_{i,\text{out}} = 1, \quad \forall F_i \in \mathcal{F}. \tag{1}$$

Meanwhile, each flow port can serve exclusively as an inlet or outlet within a flow path to ensure the consistency of the flow direction.

$$q^n_{i,\text{in}} + q^n_{i,\text{out}} \le 1, \quad \forall F_i \in \mathcal{F},\ n \in \mathcal{P}. \tag{2}$$

Further, chip design may feature more flow ports than required for a bioassay. This surplus demands a binding to assign inlets and outlets to the available flow ports.

$$\sum_{n\in\mathcal{P}} q^n_{io} = 1, \quad \forall io \in \mathcal{B}, \quad \sum_{io\in\mathcal{B}} q^n_{io} \le 1, \quad \forall n \in \mathcal{P}. \tag{3}$$

In particular, once a flow port is bound to an inlet or outlet within $\mathcal{B}$, it must consistently retain this role in any flow path in which it participates. For example, to avoid a flow port n bound to an inlet io from serving as an outlet in any flow path, we formulate the following constraint using the *big M method* [17] as

$$\sum_{F_i\in\mathcal{F}} q^n_{i,\text{out}} \le \left(1 - q^n_{io}\right)\cdot \varepsilon_M, \quad \forall n \in \mathcal{P}, \tag{4}$$

Table 1: Model variables

Binary variables	
$q^n_{i,\text{in}}$, $q^n_{i,\text{out}}$	Flow port n serves as an inlet or an outlet in P_i.
q^n_{io}	Flow port n is bound to $io \in \mathcal{B}$.
q^n_i, $q^{e_{(n,n')}}_i$	Vertex n or edge $e_{(n,n')}$ is part of P_i.
$q^{n,n'}_i$	Flow direction within P_i is from n to n'.
tq^n_i, $tq^{e_{(n,n')}}_i$	Vertex n or edge $e_{(n,n')}$ is traversed by reactants or reagents within F_i.
$\bar{q}_{t^n_i,j}$, $\bar{\bar{q}}_{t^n_i,j}$, $q_{t^n_i,j}$	F_j starts after t^n_i, ends before t^n_i, or is in execution at t^n_i.
$q_{t^n_i,e_{(n,n')}}$, $q_{t^n_i,M}$	Edge $e_{(n,n')}$ or module M is executed at t^n_i.
$eq_{t^n_i,M}$	Execution states of the edges within M at t^n_i.
$ioq_{t^n_i,M}$,	Equipotential state of M's flow ports at t^n_i.
$\bar{q}_{t^n_i,e_{(m,m')}}$	None of the fluid modules containing $e_{(m,m')}$ is executed at t^n_i.
$cq_{i,j}$	F_i and F_j exhibit a complete conflict.
$cq_{(i),j}$, $cq_{i,(j)}$	F_i or F_j reaches the common vertices first, respectively, to resolve the complete conflict.
$\bar{c}q_{i,j}$	F_i and F_j exhibit a partial conflict.
$\bar{t}q^n_j$	Vertex n in P_j is not traversed by the fluid in F_j.
$oq_{(i),j}$, $oq_{i,(j)}$	O_i or F_j initiates first, respectively, to resolve the operational conflict.
Continuous variables	
st_i, et_i	Start and end times of F_i.
t^n_i	Arrival time of the fluid at vertex n within F_i.
r^n_i	Effective resistance of the microfluidic network at t^n_i.
v^n_i	Main flow velocity at t^n_i.
$v_{t^n_i,e_{(n,n')}}$	Flow velocity in edge $e_{(n,n')}$ at t^n_i.
$\Delta t_{i,(n,n')}$	Absolute difference between t^n_i and $t^{n'}_i$.
ost_i	Start time of operation O_i.
t	Bioassay completion time.

where ε_M is an extremely large auxiliary constant. Specifically, if n serves as io, i.e., $q_{io}^n = 1$, (4) limits $q_{i,\text{out}}^n = 0$ for each $F_i \in \mathcal{F}$.

3.1.2 Connection Rules. To maintain unidirectional flow within the flow path, each involved vertex should exhibit an *inflow*, except at inlets, and an *outflow*, except at outlets. For n being a device or channel branch, such conditions are formulated as

$$q_i^n \le \sum_{n' \in V_n} q_i^{n',n} \le q_i^n \cdot \varepsilon_M, \quad q_i^n \le \sum_{n' \in V_n} q_i^{n,n'} \le q_i^n \cdot \varepsilon_M, \tag{5}$$

where V_n is the set of vertices directly connected to n. Specifically, if n is included in P_i, i.e., $q_i^n = 1$, (5) ensures the sums are ≥ 1, limiting at least one inflow to and outflow from n; otherwise, these sums equal 0, limiting the absence of any inflow or outflow at n. As for n being a flow port, the constraints can be formulated analogously. Meanwhile, an edge $e_{(n,n')}$ is included in P_i if and only if the flow direction is from n to n' or from n' to n.

$$q_i^{e_{(n,n')}} = q_i^{n,n'} + q_i^{n',n}, \quad \forall n \in \mathcal{N},\ n' \in V_n. \tag{6}$$

3.1.3 Identifying Vertices and Edges Traversed by Reactants or Reagents. Further, we formulate the following criteria to identify vertices and edges traversed by reactants or reagents for subsequent flow velocity calculation. If reactants or reagents traverse $n \in \mathcal{N}$, and there is a connected vertex $n' \in V_n$ with a flow direction from n to n', then n' and the corresponding edge $e_{(n,n')}$ must also be traversed by the reactants or reagents.

$$tq_i^{e_{(n,n')}} \ge tq_i^n + q_i^{n,n'} - 1, \quad tq_i^{n'} \ge tq_i^n + q_i^{n,n'} - 1. \tag{7}$$

Moreover, except for the destination vertex of F_i, at which fluid movement terminates, no other devices should inadvertently receive reactants or reagents to prevent contamination.

$$q_i^{n,n'} \le 1 - tq_i^n, \quad \forall n \in \mathcal{N},\ n' \in V_n \cap \left(\mathcal{D} - \left\{D_i^{\text{dest}}\right\}\right), \tag{8}$$

where D_i^{dest} denotes the destination vertices of F_i.

3.2 Dynamic Flow Velocity Calculation

We assume that the flow velocity remains uniform along each channel segment to simplify the flow velocity calculation.

3.2.1 Identifying Non-Serial Connections. As illustrated in Section 1, the parallel execution of multiple fluid transportations may introduce non-serial connections. Meanwhile, under the assumption mentioned above, flow velocity transitions within F_i can only occur at the arrival time of the fluid at a vertex n included in P_i. This arrival time is denoted by t_i^n. Consequently, to identify the non-serial connections, it is crucial to determine whether any other fluid transportation F_j is also in execution at t_i^n. Since F_j must be in one of the following three mutually exclusive states: starting after t_i^n, completing before t_i^n, or being in execution at t_i^n, its states can be formulated as

$$\bar{q}_{t_i^n,j} + \bar{\bar{q}}_{t_i^n,j} + q_{t_i^n,j} = 1. \tag{9}$$

For example, we characterize scenario F_j in execution at t_i^n as

$$st_j \le t_i^n + \left(1 - q_{t_i^n,j}\right) \cdot \varepsilon_M, \quad t_i^n \le et_j + \left(1 - q_{t_i^n,j}\right) \cdot \varepsilon_M. \tag{10}$$

Then, whether an edge $e_{(n,n')}$ is in execution at t_i^n can be identified as follows:

$$\sum_{F_j \in \mathcal{F}} q_{t_i^n,j} \cdot q_j^{e_{(n,n')}} \le q_{t_i^n,e_{(n,n')}} \cdot \varepsilon_M, \quad \forall n \in \mathcal{N},\ n' \in V_n. \tag{11}$$

Specifically, if the sum is ≥ 1, at least one fluid transportation executes at t_i^n with its flow path containing $e_{(n,n')}$. In this case, (11) limits that $q_{t_i^n,e_{(n,n')}}$ is set to 1. The constraint indicating whether a flow port serves as an inlet or outlet at t_i^n can be formulated analogously.

After identifying the executing edges and confirming the equipotential conditions of the flow ports at t_i^n, non-serial connections are identified by evaluating the states of modules.

$$q_{t_i^n,M} = eq_{t_i^n,M} \cdot ioq_{t_i^n,M}, \quad \forall M \in \mathcal{M}, \tag{12}$$

where $\mathcal{M}$ is a set of all modules. Specifically, at time t_i^n, $eq_{t_i^n,M}$ confirms the execution states of edges within M, and $ioq_{t_i^n,M}$ indicates whether the equipotential states of M's flow ports are satisfied. These variables can be characterized using the binary variables introduced above. For example, edges e_1, e_2, e_3, and e_4 are in execution at time t_i^n, then $eq_{t_i^n,UM_1^1} = 1$, where UM_1^1 is shown in Figure 5(b). Meanwhile, if io_1, io_2, and io_3 serve as inlets (or outlets) at t_i^n, then $ioq_{t_i^n,UM_1^1} = 1$. On the other hand, if some of io_1, io_2, or io_3 serve as inlets while others serve as outlets at t_i^n, UM_1^1 will not be identified as being in execution. For modules without flow ports, such as M_2 shown in Figure 5(a), $ioq_{t_i^n,M_2}$ is set to 1.

3.2.2 Modeling Flow Dynamics. Based on our module construction algorithm in Section 2.2, the non-serial connections will be identified as modules and connected externally in serial. Thus, the effective hydraulic resistance of the microfluidic network at t_i^n comprises the resistance contributions from modules and edges outside the modules executed at that time.

$$r_{t_i^n} = \sum_{e_{(m,m')} \in \mathcal{E}} r_{e_{(m,m')}} \cdot q_{t_i^n,e_{(m,m')}} \cdot \bar{q}_{t_i^n,e_{(m,m')}} + \sum_{M \in \mathcal{M}} r_M q_{t_i^n,M}, \tag{13}$$

where constants $r_{e_{(m,m')}}$ and r_M denote the effective resistances of $e_{(m,m')}$ and M, respectively. Here, binary variable $\bar{q}_{t_i^n,e_{(m,m')}} = 1$ indicates that none of the modules containing $e_{(m,m')}$ are executed at t_i^n, which can be formulated as

$$\begin{aligned} &\bar{q}_{t_i^n,e_{(m,m')}} \le \left(1 - q_{t_i^n,M}\right), \qquad \forall M \in \mathcal{M}_{e_{(m,m')}} \\ &\bar{q}_{t_i^n,e_{(m,m')}} \ge \sum_{M \in \mathcal{M}_{e_{(m,m')}}} \left(1 - q_{t_i^n,M}\right) - |M_{e_{(m,m')}}| + 1, \end{aligned} \tag{14}$$

where $\mathcal{M}_{e_{(m,m')}}$ is the set of modules containing $e_{(m,m')}$ and $|M_{e_{(m,m')}}|$ denotes its cardinality. After that, the *main flow velocity* at t_i^n is governed by the following constraint [14].

$$v_{t_i^n} \cdot r_{t_i^n} = \frac{\Delta p}{hw}, \tag{15}$$

where Δp denotes the input pressure, and h and w denote the height and width of the flow channel, respectively. Accordingly, the flow velocity on $e_{(m,m')}$ can be calculated as

$$v_{t_i^n}^{e_{(m,m')}} = \sum_{M \in \mathcal{M}_{e_{(m,m')}}} \alpha_M^{e_{(m,m')}} \cdot v_{t_i^n} \cdot q_{t_i^n,M} + v_{t_i^n} \cdot \bar{q}_{t_i^n,e_{(m,m')}}, \tag{16}$$

where constant $\alpha_M^{e_{(m,m')}}$ is the flow distribution ratio of $e_{(m,m')}$ within M.

Finally, according to the principle that the product of time and velocity equals distance, the fluid transit time between any vertex $n \in \mathcal{N}$ and its directly connected vertex $n' \in V_n$ must satisfy

$$l_{e_{(n,n')}} = v_{t_i^n, e_{(n,n')}} \cdot \Delta t_{i,(n,n')}. \tag{17}$$

Here, continuous variable $\Delta t_{i,(n,n')}$ denotes the absolute difference between t_i^n and $t_i^{n'}$, which can be formulated as

$$\Delta t_{i,(n,n')} \geq t_i^n - t_i^{n'}, \quad \Delta t_{i,(n,n')} \geq t_i^{n'} - t_i^n. \tag{18}$$

As minimizing the bioassay completion time is our optimization objective, $\Delta t_{i,(n,n')}$ is constrained to take the larger value of two right-hand side terms.

3.3 Conflict Identification and Resolution

We classify conflicts between two fluid transportations into two types: *complete*, where distinct fluids encounter at the same vertices, and *partial*, where the execution of one fluid transportation obstructs another's access to its inlet or outlet. Meanwhile, we define *operational conflict* as when an operation occupies a device necessary for executing a fluid transportation. The following details the identification and resolution of conflicts.

3.3.1 Complete Conflict. As illustrated in Section 1, F_2 and F_3 in Figure 1 exhibit a complete conflict due to the transportation of distinct fluids along a shared subpath $s_2 \rightarrow s_3$. Specifically, a complete conflict arises when fluids within two transportations, F_i and F_j, traverse at least one common vertex, resulting in the risk of contamination.

$$\sum_{n \in \mathcal{N}} tq_i^n \cdot tq_j^n \leq cq_{i,j} \cdot \varepsilon_M, \quad \forall F_i \in \mathcal{F},\ F_j \in \mathcal{F}_i^c, \tag{19}$$

where $\mathcal{F}_i^c$ is the set of fluid transportations that transport different fluids from F_i, excluding cases where F_j and F_i transport mixing-required fluids. If F_i and F_j exhibit a complete conflict, the fluid within F_i or F_j should reach the common vertices first to resolve the conflict.

$$cq_{(i),j} + cq_{i,(j)} = cq_{i,j}, \tag{20}$$

where $cq_{(i),j}$ and $cq_{i,(j)}$ are binary variables indicating which fluid reaches the common vertices first: $cq_{(i),j} = 1$ means that F_i takes priority, while $cq_{i,(j)} = 1$ indicates the opposite. For example, the scenario where F_i reaches the common vertices first is formulated as

$$t_j^n \geq t_i^n - \left(2 - cq_{(i),j} - tq_i^n \cdot tq_j^n\right) \cdot \varepsilon_M, \quad \forall n \in \mathcal{N}. \tag{21}$$

Specifically, if $n \in \mathcal{N}$ is one of the vertices shared by F_i and F_j, i.e., $tq_i^n \cdot tq_j^n = 1$, and F_i is set to reach the common vertices first, i.e., $cq_{(i),j} = 1$, then (21) ensures that F_j arrives at n after F_i.

3.3.2 Partial Conflict. As illustrated in Section 1, F_1 and F_2 in Figure 1 demonstrate a partial conflict because the execution of F_1 requires blocking a subpath within P_2, which disrupts the necessary pressure for fluid movement in F_2. This occurs because there are critical vertices shared between P_1 and P_2, which are traversed by fluids within F_1 but not by fluids within F_2; instead, they serve as connections to the inlet of P_2. The scenario of executing F_i disrupting the connection of F_j can be characterized as

$$\sum_{n \in \mathcal{N}} tq_i^n \cdot \bar{t}q_j^n \leq \bar{c}q_{i,j} \cdot \varepsilon_M, \quad \forall F_i,\ F_j \in \mathcal{F}, \tag{22}$$

where binary variable $\bar{t}q_j^n = 1$ indicates that vertex n in P_j is not traversed by fluids within F_j, and can be formulated as

$$\bar{t}q_j^n \leq q_j^n, \quad \bar{t}q_j^n \leq 1 - tq_j^n, \quad \bar{t}q_j^n \geq q_j^n - tq_j^n. \tag{23}$$

Similar to the resolution of a complete conflict, addressing a partial conflict involves exclusively ensuring that the fluid within F_i reaches the critical vertices first or permitting F_j to finish its transportation ahead. The linear constraints that describe resolving partial conflicts are formulated analogously to (20) and (21).

3.3.3 Operational conflict. Suppose an operation O_i is bound to a device D_k based on the binding function. An operational conflict occurs if fluid transportation F_j also requires the same device, i.e., $q_j^{D_k} = 1$. The initiation of O_i or F_j must be postponed until the other completes to resolve this conflict.

$$oq_{(i),j} + oq_{i,(j)} = q_j^{D_k}, \tag{24}$$

where binary variables $oq_{(i),j}$ and $oq_{i,(j)}$ indicate whether O_i or F_j initiates first, respectively. For example, O_i initiating first can be formulated as

$$st_j \geq (ost_i + c_i) - \left(1 - oq_{(i),j}\right) \cdot \varepsilon_M. \tag{25}$$

Finally, the objective function is set to minimize the bioassay completion time, represented by a continuous variable t. To this end, the last constraint is introduced

$$\forall_{F_i \in \mathcal{F}}: \quad t \geq et_i, \tag{26}$$

and the overall problem is modeled as

minimize t

Subject to (1)–(26)

4 EXPERIMENTAL RESULTS

This section investigates the performance of the proposed method using four chip designs as test cases. Cases 1 and 4 are proposed as benchmarks in [4]. Case 2 is a synthetic benchmark created with Columba 2.0 [5]. Case 3 is an application-specific design proposed in [5]. Our work was implemented using C++, and the optimizations were run on a computer with a 1.60 GHz CPU. The QP model is solved by Gurobi [18]. Gurobi solves the QP model [18]. In our experimental setup, the input pressure Δp was set to 100 Pa, with the channel dimensions set to 50 µm in height and 100 µm in width. Additionally, each operation was assigned an execution time of 2 seconds.

We compare the proposed method with the state-of-the-art approach in [4], known as VOM. Specifically, VOM constructs valid flow paths for given chip designs with adjustable optimization criteria, including execution time and resource usage. In our comparative analysis, we utilized VOM's execution time optimization criteria to construct flow paths. We addressed its limitations in conflict detection and resolution by generating scheduling schemes by sequentially executing fluid transportations.

Table 2: Test cases used in the experiments and comparison of results.

Case	$\|\mathcal{F}\|$	$\|\mathcal{O}\|$	$\|\mathcal{B}\|$	$\|\mathcal{D}\|$	$\|\mathcal{P}\|$	$\|\mathcal{S}\|$	$\|\mathcal{E}\|$	$\|\mathcal{M}\|$	Bioassay completion time (s)		Improvement (%)	Runtime (s)
									Proposed method	VOM		
1	4	2	3	2	4	2	7	51	50.2	57.4	12.5	1.7
2	6	2	4	3	5	3	10	1251	82.6	156.5	47.2	162.1
3	8	4	4	4	5	9	21	143	89.2	147.2	39.4	13.5
4	9	3	5	3	5	3	9	1242	81.2	228.9	64.5	1456.3

$|\mathcal{F}|$: the number of fluid transportations; $|\mathcal{O}|$: the number of operations; $|\mathcal{B}|$: the number of specified inlets and outlets; $|\mathcal{D}|$: the number of devices; $|\mathcal{P}|$: the number of flow ports; $|\mathcal{S}|$: the number of channel branches; $|\mathcal{E}|$: the number of channel segments; $|\mathcal{M}|$: the number of fluid modules.

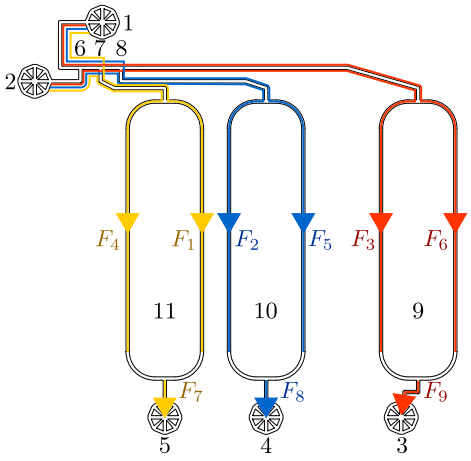


Figure 6: Illustration of the flow-layer structure and the required fluid transportations used in case 4, where vertices 1–5 are flow ports, vertices 6–8 are channel branches, and vertices 9–11 are three mixers.

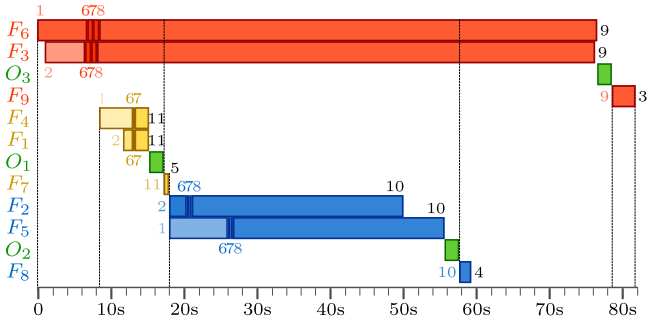


Figure 7: Illustration of the optimized scheduling scheme of case 04: The x-axis represents time in seconds, while the y-axis represents fluid transportations and operations. The execution duration of each fluid transportation is shown as a series of rectangles, where each rectangle represents the duration of fluid movement between two consecutive vertices along the corresponding flow path. The labels above each rectangle indicate the starting and the destination vertices. The color intensity of the rectangles indicates flow velocity, with darker colors representing higher values. Green rectangles represent the duration of operations.

4.1 Bioassay Completion Time Comparison

Table 2 showcases the input features and contrasts the bioassay completion time using the proposed method against VOM. This comparison reveals that parallel execution of fluid transportations significantly shortens bioassay durations, with the proposed method achieving an average time reduction of 40.9% compared to VOM. Generally, the number of required fluid transportations and operations indicates the optimization space. However, despite case 3 having a higher demand, the reduction in completion time is less pronounced than in case 2. This discrepancy can be attributed to the limitations in parallel execution resulting from conflict avoidance requirements. Further, observation indicates that although case 3 contains the highest number of vertices and edges, it features far fewer modules than cases 2 and 4. In other words, the presence of non-serial connections is topological and not directly proportional to the number of vertices and edges.

4.2 Case Study

We illustrate the flow-layer structure and the required fluid transportations used in case 4 in Figure 6, and the optimized scheduling scheme in Figure 7. The following are key observations regarding conflict identification and resolution:

- F_3 and F_6 transport mixing-required fluids and are executed in parallel without contamination risk, as are F_2 with F_5 and F_1 with F_4.
- F_3/F_6, F_1/F_4, and F_2/F_5 each exhibit a complete conflict with the others at common vertices 6 and 7, or at critical vertices 6, 7, and 8. Therefore, the fluids within them traverse these critical vertices at different times to avoid conflicts.
- F_7 has a partial conflict with F_3/F_6 and F_2/F_5 due to the critical vertex 7. During execution, no conflict occurs with F_3/F_6 since fluids within F_3/F_6 already traverse vertex 7 when F_7 starts, while F_2/F_5 initiate after F_7 completes to avoid conflicts. Similarly, F_8 starts after the fluids within F_3/F_6 traverse critical vertex 8, thus avoiding the partial conflict.

Further, a comparison of fluid movement duration between vertices 6 and 7 within different fluid transportations shows that fluids traverse these vertices at different flow velocities. Specifically, the flow velocity on edge $e_{(6,7)}$ varies depending on the concurrent executions: it is 16.4 mm s^{-1} during the parallel execution of F_3 and F_6, increases to 65.1 mm s^{-1} during $F_3 \circ F_6 \circ F_1 \circ F_4$, and reaches 37.7 mm s^{-1} during $F_3 \circ F_6 \circ F_2 \circ F_5$. This confirms the dynamic influence of channel connectivity on flow velocities.

5 CONCLUSION

In this work, we proposed a dynamic topology-aware high-level synthesis method to realize the bioassay with minimized completion time. To this end, the proposed method integrates a flow velocity model that accurately calculates flow velocities considering the interactions among concurrently executed fluid transportations and operations. The proposed method was implemented by constructing a QP model that identifies and resolves conflicts during the parallel execution of multiple fluid transportations. Experimental results confirmed that the proposed method could construct valid flow paths and optimize scheduling schemes to execute fluid transportations in parallel without risk of contamination and achieve the minimized bioassay completion time.

ACKNOWLEDGMENTS

This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project Number 515003344.

REFERENCES

[1] Jong Wook Hong, Vincent Studer, Giao Hang, W French Anderson, and Stephen R Quake. A nanoliter-scale nucleic acid processor with parallel architecture. *Nature Biotechnology*, 22(4):435–439, 2004.

[2] Bidhan Chandra Dhar and Nae Yoon Lee. Lab-on-a-chip technology for environmental monitoring of microorganisms. *BioChip Journal*, 12(3):173–183, 2018.

[3] Wajid Hassan Minhass, Paul Pop, Jan Madsen, Mette Hemmingsen, and Martin Dufva. System-level modeling and simulation of the cell culture microfluidic biochip procell. In *2010 Symposium on Design Test Integration and Packaging of MEMS/MOEMS (DTIP)*, pages 91–98, 2010.

[4] Mengchu Li, Tsun-Ming Tseng, Yanlu Ma, Tsung-Yi Ho, and Ulf Schlichtmann. Vom: Flow-path validation and control-sequence optimization for multilayered continuous-flow microfluidic biochips. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8, 2019.

[5] Tsun-Ming Tseng, Mengchu Li, Daniel Nestor Freitas, Travis McAuley, Bing Li, Tsung-Yi Ho, Ismail Emre Araci, and Ulf Schlichtmann. Columba 2.0: A co-layout synthesis tool for continuous-flow microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(8):1588–1601, 2018.

[6] Kai-Han Tseng, Sheng-Chi You, Jhe-Yu Liou, and Tsung-Yi Ho. A top-down synthesis methodology for flow-based microfluidic biochips considering valve-switching minimization. In *Proceedings of the 2013 ACM International Symposium on Physical Design*, ISPD '13, pages 123–129, New York, NY, USA, 2013. Association for Computing Machinery.

[7] Hailong Yao, Qin Wang, Yizhong Ru, Yici Cai, and Tsung-Yi Ho. Integrated flow-control codesign methodology for flow-based microfluidic biochips. *IEEE Design & Test*, 32(6):60–68, 2015.

[8] Mengchu Li, Tsun-Ming Tseng, Bing Li, Tsung-Yi Ho, and Ulf Schlichtmann. Sieve-valve-aware synthesis of flow-based microfluidic biochips considering specific biological execution limitations. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 624–629, 2016.

[9] Mengchu Li, Tsun-Ming Tseng, Bing Li, Tsung-Yi Ho, and Ulf Schlichtmann. Component-oriented high-level synthesis for continuous-flow microfluidics considering hybrid-scheduling. In *Proceedings of the 54th Annual Design Automation Conference 2017*, DAC '17, New York, NY, USA, 2017. Association for Computing Machinery.

[10] Wajid Hassan Minhass, Paul Pop, and Jan Madsen. System-level modeling and synthesis of flow-based microfluidic biochips. In *2011 Proceedings of the 14th International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES)*, pages 225–233, 2011.

[11] Wajid Hassan Minhass, Jeffrey McDaniel, Michael Raagaard, Philip Brisk, Paul Pop, and Jan Madsen. Scheduling and fluid routing for flow-based microfluidic laboratories-on-a-chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(3):615–628, 2018.

[12] Tsun-Ming Tseng, Mengchu Li, Bing Li, Tsung-Yi Ho, and Ulf Schlichtmann. Columba: co-layout synthesis for continuous-flow microfluidic biochips. In *Proceedings of the 53rd Annual Design Automation Conference*, DAC '16, New York, NY, USA, 2016. Association for Computing Machinery.

[13] Tsun-Ming Tseng, Mengchu Li, Daniel Nestor Freitas, Amy Mongersun, Ismail Emre Araci, Tsung-Yi Ho, and Ulf Schlichtmann. Columba s: A scalable co-layout design automation tool for microfluidic large-scale integration. In *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, pages 1–6, 2018.

[14] Kwang W. Oh, Kangsun Lee, Byungwook Ahn, and Edward P. Furlani. Design of pressure-driven microfluidic networks using electric circuit analogy. *Lab Chip*, 12:515–545, 2012.

[15] Krishnendu Chakrabarty and Jun Zeng. Design automation for microfluidics-based biochips. *J. Emerg. Technol. Comput. Syst.*, 1(3):186–223, oct 2005.

[16] Béla Bollobás. *Modern Graph Theory*, pages 1–37. Springer New York, New York, NY, 1998.

[17] Igor Griva, Stephen G Nash, and Ariela Sofer. *Linear and Nonlinear Optimization: Second Edition*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2009.

[18] Gurobi Optimization LLC. *Gurobi Optimizer Reference Manual*, 2022.